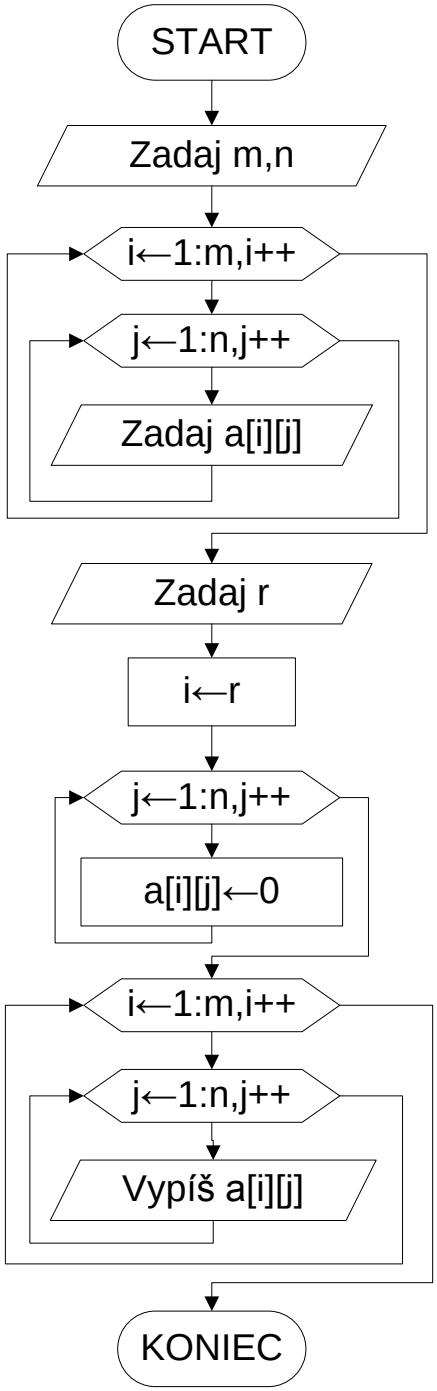
9. Algoritmy s indexovanou premennou – dvojrozmerné pole

1. Vytvorte algoritmus, ktorý načíta prvky do matice a následne prvky v jednom z riadkov zadanom používateľom zmení hodnotu všetkých prvkov na 0.

   Vstupné premenné: m, n, a[i][j], r
   Pomocné premenné: i, j
   Výstupná premenná: a[i][j]

   Analýza riešenia:
   Po zadaní počtu riadkov a stĺpcov použitím dvoch cyklov FOR s indexovaním i pre riadky a j pre stĺpce zadáme prvky matice. Následne zadáme číslo riadku (od 1 do m), v ktorom chceme všetky prvky zmeniť na nuly. Zadané číslo riadku načítame do premennej r a pre zmenu prvkov riadku nepoužijeme 2 cykly FOR ako pri načítavaní, pretože nemeníme hodnoty prvkov vo všetkých riadkoch. Preto premennej i priradíme hodnotu premennej r a cyklus FOR použijeme len pre zmenu indexov stĺpcov, v ktorých má byť zmena vykonaná, teda budeme meniť hodnoty prvkov: a[r][1], a[r][2],..., a[r][n]. Následne vypíšeme všetky prvky matice po zmene zadaného riadku.

   Slovný popis algoritmu:
   1. krok: zadaj počet riadkov m, počet stĺpcov n
   2. krok: ak i=1 až m, pokračuj krokom 3, inak prejdi na krok 5
   3. krok: ak j=1 až n, pokračuj krokom 4, inak sa vráť na krok 2
   4. krok: zadaj a[i][j], vráť sa na krok 3
   5. krok: zadaj číslo riadku r, ktorý chceš meniť
   6. krok: premennej i priraď hodnotu r
   7. krok: ak j=1 až n, pokračuj krokom 8, inak prejdi na krok 9
   8. krok: premennej a[i][j] priraď hodnotu 0, vráť sa na krok 7
   9. krok: ak i=1 až m, pokračuj krokom 10, ukonči program
   10. krok: ak j=1 až n, pokračuj krokom 11, inak sa vráť na krok 9
   11. krok: vypíš a[i][j], vráť sa na krok 10



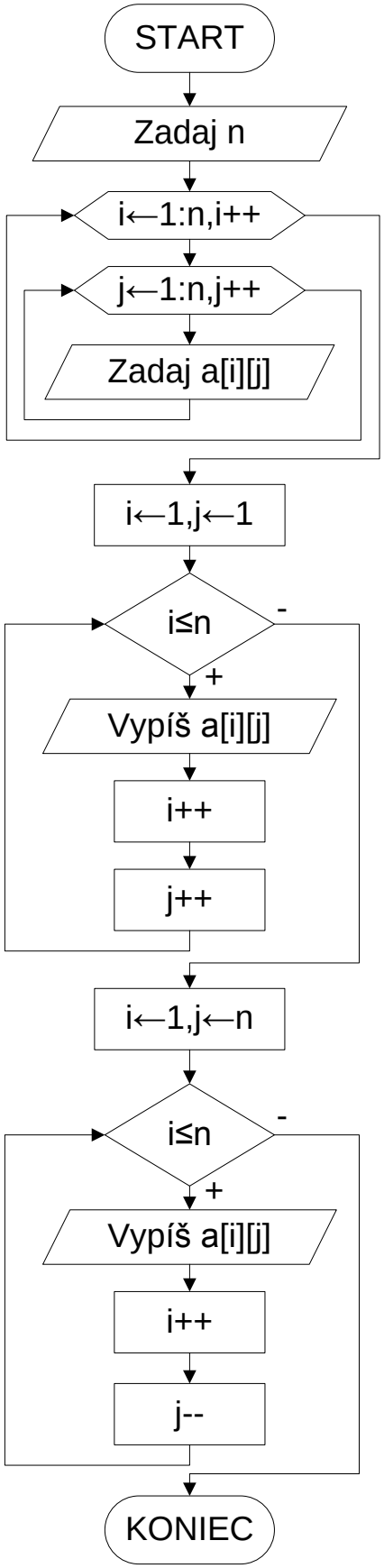

```
int main()
{
 int m,n,r,i,j,a[i][j];

 printf("Zmena prvkov zadaneho riadku na nuly\n\n");

 printf("Zadaj pocet riadkov m: ");
 scanf("%d",&m);

 printf("Zadaj pocet stlpcov n: ");
 scanf("%d",&n);

 int a[m][n];

 printf("\n");

 for(i=1;i<=m;i++){
        for(j=1;j<=n;j++){
                printf("Zadaj cislo a[%d][%d]: ",i,j);
                scanf("%d",&a[i][j]);
                }
        }

 printf("\nZadaj riadok, ktory chces zmenit na 0: ");
 scanf("%d",&r);
 i=r;
 for(j=1;j<=n;j++){
        a[i][j]=0;
        }

 for(i=1;i<=m;i++){
        printf("\n");
        for(j=1;j<=n;j++){
                printf("%d  ",a[i][j]);
                }
        }
 printf("\n\n");
 system("PAUSE");
 return 0;
}
```

2. Vytvorte algoritmus, ktorý vypíše hlavnú a vedľajšiu diagonálu zadanej štvorcovej matice, pričom používateľ na vstupe nezadá počet riadkov aj stĺpcov, ale len jedno číslo, ktoré bude symbolizovať počet riadkov, ako aj stĺpcov.

Vstupné premenné: n, a[i][j]
Pomocné premenné: i, j
Výstupná premenná: a[i][j]

Analýza riešenia:
Zadáme jednotlivé prvky matice. Hlavná diagonála začína vždy prvkom a[1][1], pokračuje a[2][2], ..., a končí prvkom a[n][n]. To znamená, že hodnota indexu j je rovnaká ako hodnota indexu i. Preto najprv nastavíme hodnotu oboch premenných na 1 a cyklom WHILE overíme, či je hodnota ľubovoľného z nich menšia alebo rovná hodnote n. Ak podmienka v cykle WHILE je splnená, vypíšeme prvok a[i][j] čo je v prvom prípade a[1][1], následne zvýšime hodnotu oboch indexových premenných o 1, čím a opakujeme cyklus až kým program nevypíše hodnotu prvku a[n][n]. Pri vypisovaní vedľajšej diagonály je postup takmer rovnaký až nato, že prvý prvok je a[1][n] a posledný a[n][1]. Čo znamená, že zatiaľ čo hodnotu premennej i pri každom behu cyklu zvyšujeme o 1, premennú j naopak o 1 znižujeme, pričom sme pred zbehnutím cyklu nastavili hodnotu hodnoty premenných i a j nasledovne: i=1, j=n.

Slovný popis algoritmu:
1. krok: zadaj počet riadkov a stĺpcov n
2. krok: ak i=1 až n, pokračuj krokom 3, inak prejdi na krok 5
3. krok: ak j=1 až n, pokračuj krokom 4, inak sa vráť na krok 2
4. krok: zadaj a[i][j], vráť sa na krok 3
5. krok: nastav i=1, j=1
6. krok: kým i≤n, opakuj kroky 7-9, inak prejdi na krok 10
7. krok: vypíš a[i][j]
8. krok: i zvýš o 1
9. krok: j zvýš o 1, vráť sa na krok 6
10. krok: nastav i=1, j=n
11. krok: kým i≤n, opakuj kroky 12-14, inak ukonči program
12. krok: vypíš a[i][j]
13. krok: i zvýš o 1
14. krok: j zníž o 1, vráť sa na krok 11




```
int main()
{
 int n,i,j,a[i][j];

 printf("Zadaj pocet riadkov a stlpcov n: ");
 scanf("%d",&n);

 printf("\n");

 for(i=1;i<=n;i++){
        for(j=1;j<=n;j++){
                printf("Zadaj cislo a[%d][%d]: ",i,j);
                scanf("%d",&a[i][j]);
                }
        }

 i=1;
 j=1;
 printf("\nHlavna diagonala obsahuje prvky: ");
 while(i<=n){
        printf("%d ",a[i][j]);
        i++;
        j++;
        }

 i=1;
 j=n;
 printf("\nVedlajsia diagonala obsahuje prvky: ");
 while(i<=n){
        printf("%d ",a[i][j]);
        i++;
        j--;
        }

 printf("\n\n");
 system("PAUSE");
 return 0;
}
```